

Serial No. 09/387,654

Appeal Brief

Page 1 of 30



P A T E N T

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: Michel K. Bowman-Amuah

Serial No: 09/387,654

Filed: August 31, 1999

Title: TRANSACTION SERVICES  
PATTERNS IN A NETCENTRICK  
ENVIRONMENT

Group Art Unit: 3629

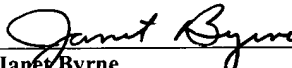
Examiner: M. Fisher

Docket No: 60021-327501

MS Appeal Brief - Patents  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

CERTIFICATE OF TRANSMISSION

I certify that this correspondence is being deposited with the U.S. Postal Service as First Class Mail with sufficient postage and is addressed to Mail Stop Appeal Brief - Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the date shown below.

  
Janet Byrne

7-7-06  
Date

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE  
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

APPELLANT'S APPEAL BRIEF

Dear Sir:

In response to the Office Action dated November 18, 2005, which prompted Appellant's Notice of Appeal of February 17, 2006, Appellant appeals the rejections of Examiner Fisher and submits this Appeal Brief.

07/10/2006 CNEGAI 00000018 501901 05387654

01 FC:1402 500.00 DA

**(1) REAL PARTY IN INTEREST**

The present patent application has been assigned to Accenture LLP, an Illinois corporation.

**(2) RELATED APPEALS AND INTERFERENCES**

There are no appeals or interferences related to the present case.

**(3) STATUS OF CLAIMS**

Claims 1-18 are pending and appealed. Claims 1-18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Grewal (U.S. Patent No. 5,592,672) in view of Nordstrom (U.S. Patent No. 6,085,277).

**(4) STATUS OF AMENDMENTS**

No claim amendments were filed subsequent to the final Office Action.

**(5) SUMMARY OF CLAIMED SUBJECT MATTER**

The present patent application relates to transaction service patterns in a net-centric environment. Briefly, a transaction is a set of business events that, coupled together, accomplish a particular business function, such as turning on a gas service. The business events are logically related so their data changes to stored data are logically related as well. Taken together, these data changes create a new, consistent state for the business model. The state of the business model may not be consistent while the transaction is in progress and so it becomes desirable to manage the entire transaction from its point of origin to its point of completion. To help manage the transaction, each transaction is organized through a logical unit of work. The logical unit of work manages the business model and any of its data changes. The business model corresponds with a set of business objects. In the process of managing its business model, the logical unit of work will often send messages to all business objects with the logical unit of work.

The above-described system also uses patterns to address at least two problems related to managing the transactions. A problem can occur if the business models of concurrent logical units of work overlap and transactions attempt to write to the same business object. An example of this is a call center representative trying to solve more than one customer problem during the course of one call. This problem is addressed with a pattern in this application of assigning each logical unit of work independent copies of the relevant portions of the business model. Another problem can occur when too many network messages are sent, which can overwhelm a network. One example where this problem can arise is when one request may depend on another request such as when a business transaction that tries to retrieve a customer ID might also want to retrieve a customer's address ID. The transaction will not have the address ID, however, until the customer ID is retrieved. Accordingly, multiple messages are sent when one request is dependent on another. This problem is addressed with a pattern that allows a batched request to indicate that it depends on another request.

Independent claims 1, 7, and 13 recite a method, a computer program embodied on a computer-readable medium, and a computer-readable storage medium containing a set of instructions for a general purpose computer, respectively, for implementing transaction service patterns. The transaction service patterns include:

(a) batching logically related requests for reducing network traffic (page 694, lines 4-5, Fig. 185, and generally the discussion of Figs. 186, 187 on pages 694), including the steps of managing a group of business objects necessary for a transaction in a logical unit of work (page 687, lines 4-6, Fig. 182), and grouping the logically related requests received from the logical unit of work into a single network message (page 694, lines 4-9, Fig. 185), wherein the logically related requests include at least a dependent batched request and a parent batched request (*Id.* and page 683, lines 13-17.);

(b) indicating whether the dependent batched request depends on the response to the parent batched request, including the steps of receiving a register that the dependent batched request is dependent upon response data from the parent batched request, receiving a response to the parent request, directing data from the response to the parent request to the dependent batched request; and receiving a response to the dependent batched request based on the response to the parent request (page 683, lines 10-25, Fig. 179, and generally the discussion of Figs. 180-181 on pages 685-686);

(c) sending the single network message to the group of business objects necessary for the logical unit of work (page 687, lines 3-10, Fig. 182);

(d) sorting the logically related requests that are unbatched from a batched message (page 696, line 19 to page 697, line 18, Fig. 188); and

(e) providing multiple logical units of work operating concurrently, wherein the logical unit of work is one of the multiple logical units of work, such that each of the multiple logical units of work manipulates at least one of the group of business objects that is common to each of the multiple logical units of work, including the steps of creating a copy of the common business object for each of the logical units of work such that the copy of the common business object for each of the logical units of work is a separate instance of the common business object, and verifying that a change to one instance of the common business object does not change the other copies of the common business object (page 701, line 14 to page 702, line 6 and Fig. 191; and generally the discussion of Figs. 192-193 on pages 702-705).

**(6) GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL**

Claims 1-18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Grewal (U.S. Patent No. 5,592,672) in view of Nordstrom (U.S. Patent No. 6,085,277). In making this obviousness rejection, has the Patent Office established a *prima facie* case of obviousness?



## **(7) ARGUMENT**

### **A. Background of the Prosecution**

Appellants filed the original patent application on August 31, 1999. The application included 18 claims of which 3 claims were independent claims. Upon Appellants petition, the Patent Office decided to make the application special on August 21, 2000.

The first substantive Office Action addressing the merits of patentability was mailed on October 8, 2003. This Office Action rejected all 18 claims. Specifically, claims 13-18 were rejected under 35 U.S.C. 101; claims 13-18 were rejected under 35 U.S.C. 112; and claims 1-18 were rejected under 35 U.S.C. 103(a) over Grewal (U.S. Patent No. 5,592,672). Appellants responded with an Amendment on January 8, 2004, and amended claims 1-18.

A First Final Office Action was mailed on March 29, 2004. The First Final Office Action removed the previous rejections under 35 U.S.C. 101, 112, but rejected the claims 1-18 under 35 U.S.C. 103(a) over Grewal (U.S. Patent No. 5,592,672) in view of Nordstrom (U.S. Patent No. 5,592,672).

Appellants filed a First Response After Final on May 6, 2004, requesting reconsideration of the finality of the First Final Office Action. In general, Appellants argued that the subject limitations of the independent claims were relocated subject matter from the dependent claims, and these amendments did not necessitate a new rejection. Rather, it was the shortcoming of the Grewal reference that necessitated the new rejection.

In reply to the First Response After Final, a Second Final Office Action was mailed on September 1, 2004. The Second Final Office Action did not address either the arguments of the First Response After Final of May 6, 2004, or the Amendment of January 8, 2004. Rather, the Second Final Office Action merely asserted that Appellants "arguments with respect to the claims have been considered but are moot in view of the new ground(s) of the rejection." However, there was no new ground of rejection in the Second Final Office Action – the rejections were identical to the very first Non-Final Office Action (mailed on October 8, 2003) and included rejections based on 35 U.S.C. 101, 112, and 103. These deficiencies of the Second

Final Office Action were pointed out in the Second Response After Final, filed on November 5, 2004, which also requested reconsideration of the finality of the Second Final Office Action.

An Advisory Action was mailed on December 16, 2004, stating that the request for reconsideration in the Second Response After Final had been considered but did not place the application in condition for allowance.

A Request for Continued Examination and Amendment was filed on December 29, 2004. Claims 1-18 were pending, and claims 1, 3, 7, 9 and 13-18 were amended. A Non-Final Office Action was mailed on March 28, 2005, rejecting claims 1-18.

In response to the Amendment, however, the Office Action of March 28, 2005, removed the rejections based on 35 U.S.C. 101 and 112. Claims 1-18 were rejected under 35 U.S.C. 103(a) as being unpatentable over Grewal (U.S. Patent No. 5,592,672) in view of Nordstrom (U.S. Patent No. 5,592,672).

Appellants did not amend the claims in their response to the Office Action of March 28, 2005, and instead traversed the rejections of claims 1-18 under 35 U.S.C. 103(a). A Final Office Action was mailed on November 18, 2005.

In response to the Final Office Action of November 18, 2005, Appellants filed a Notice of Appeal on February 17, 2006, and now submit this Appeal Brief.

#### **B. The Rejection of the Appealed Claims should be Reversed**

The final Office Action of November 18, 2005, i.e., the final Office Action that prompted this appeal, rejected the pending claims, i.e., claims 1-18. Claims 1-18 were rejected under 35 U.S.C. 103(a) as being unpatentable over Grewal in view of Nordstrom. The final Office Action of November 18, 2005, states that Grewal does not teach two features of the independent claims (claims 1, 7, and 13), which include:

- “verifying that the various messages do not change in response to a change of another message;” (page 2) and
- “batching the files into logically related requests so as to send a single network message with a parent batch and dependent batch” (page 2).

With regard to the former feature, the final Office Action states that it “would be obvious . . . to modify the structure of Grewal by making sure that changes to one message do not change another message to ensure that incorrect message are not sent” (page 2). The Nordstrom reference is not applied in this feature.

With regard to the latter feature, the final Office Action states “Nordstrom discloses an interrupt and message batching method that would inherently have a parent batch and dependent batch as Nordstrom teaches batching messages together.” In addition, it “would have been obvious . . . to modify the system as disclosed by Grewal by batching messages as Grewal discloses holding messages in a queue until a message number threshold is reached” (page 3).

The rejections related to both features are discussed in detail below.

1. The Rejection related to the Feature of Verifying that the Various Messages do not Change in Response to a Change of Another Message

The final Office Action states that Grewal does not teach “verifying that the various messages do not change in response to a change of another message,” as set forth in independent claims. Nordstrom is not cited as teaching this feature either. Instead, the final Office Action continues that it “would be obvious . . . to modify the structure of Grewal by making sure that changes to one message do not change another message to ensure that incorrect message are not sent.” Appellants submit that this modification to Grewal is irrelevant because the proposed modification does not correctly correspond with a limitation in the claim. Specifically, there is no limitation in the claim of “verifying that the various messages do not change in response to a change of another message.”

The limitation from the independent claims at issue here is apparently “verifying that a change to one instance of the common business object does not change the other copies of the common business object.” A distinction here being the claim limitation describes a “business object” versus another limitation in the independent claims of a “message” applied in the rejection. In making the rejection, the final Office Action unreasonably construes the limitations

of a “message” and a “business object” as the same item, rather than the correct interpretation that these two claim limitations are separate and distinct.

Appellants will demonstrate that the correct interpretation of the limitations “messages” and “business objects” in the claims is that they are separate and distinct limitations and not the same item as in the final Office Action. Appellants will demonstrate that their interpretation is consistent with the plain language of the claims, and the interpretation in the final Office Action is unreasonable and thus cannot stand. Appellants will also demonstrate that the prior art does not teach or suggest a corresponding feature of the correctly interpreted claim limitation, and thus the claims are patentably distinguishable from the prior art of record.

*a. Limitations of the Independent Claims have been Misinterpreted*

The Office Action misinterprets the limitations of “business objects” and “messages” in the independent claims. The limitation of “business objects” is in the feature of “managing a group of business objects necessary for a transaction in a logical unit of work” as set forth in element (a) in the independent claims. Additionally, the limitation of “messages” is in the feature of “grouping the logically related requests received from the logical unit of work into a single network message” as set forth in element (a) of the independent claims. Still further, element (c) of the independent claims contains the limitation of “sending the single network message to the group of business objects necessary for the logical unit of work.” Appellants assert that the limitations of “business objects” and “messages” are separate and distinct limitations of the claims.

Rather than treat the two limitations as separate and distinct, the final Office Action has treated the limitations as being the same, despite Appellants arguments to the contrary in at least the response of August 15, 2005. For example, the final Office Action rejected the claims and Appellants’ arguments because “messages could be a group of business objects as the system could be used by a business,” as set forth on page 2 of the final rejection of November 18, 2005. In addition, the Response to Argument section on page 4 of the same final rejection sets out that as “to the arguments in relation to the language of the claims, the message would contain the ‘common business object’.” The final Office Action states that the limitations are either

identical or have one incorporated within the other. In other words, the messages are or could be the business object or the messages contain the common business object. Appellants assert that this interpretation is both incorrect and unreasonable

*b. The Plain Language of the Independent Claims Compels a Different Interpretation than that in the Final Office Action*

The plain language of the independent claims compels the conclusion that the limitations of “messages” and “business objects” be interpreted as separate and distinct, rather than the same, limitations. Examples demonstrating the distinctiveness of these limitations are set forth in the independent claims, and include:

- “business objects necessary for a transaction [are grouped] into a logical unit of work, and [] the logically related requests received from the logical unit of work [are grouped] into a single network message” (element (a) in the independent claims);
- “each of the multiple logical units of work manipulates at least one of the group[s] of business objects that is common to each of the multiple logical units of work” (element (e) in the independent claims); and
- “single network message [is sent] to the group of business objects necessary for the logical unit of work” (element (c) in the independent claims).

These steps in this example clearly indicate that (1) business objects are grouped into a logical unit of work; (2) logically related requests are received from the logical unit of work; and (3) the logically related requests are grouped into a single network message. Accordingly, the limitations of “message” and “common business object” have separate functions, origins, and purposes within the plain language of the claims, and any interpretations other than that the limitations are separate and distinct are simply incorrect.

*c. The Interpretation in the Final Office Action is Unreasonable and Thus Cannot Stand*

Appellants assert that the claim interpretation of the final Office Action is not only incorrect but is also unreasonable and thus cannot stand under MPEP 2111. There is no question

of proper scope or of definiteness at issue here. Instead, Appellants assert that the incorrect interpretation of the claim limitations violates basic precepts of claim interpretation, and that such an interpretation against the plain meaning of the claim is unreasonable.

For example, logic compels the conclusion that two limitations having separate functions, origins and purpose, that are not otherwise specifically called-out to be the same, are each given their own specific meaning. This is especially true in the case of the present claims because the claims make sense under the logical conclusion that the limitations are separate and distinct. The claims do not make sense in view of the specification under the interpretation in the final Office Action. In addition, the interpretation in the final Office Action would render meaningless the otherwise clear plain-language interrelationships between the “common business objects,” the “logical units of work,” the “requests,” and the “messages.”

The only reasonable interpretation of the claims is to treat the limitations of “message and “common business objects” as separate and distinct. The interpretation in the final Office Action that they are not separate and distinct because “messages could be a group of business objects as the system could be used by a business,” and “the message would contain the ‘common business object’” is unreasonable. Accordingly, Appellants respectfully request that the Board of Appeals replace the interpretation of the claims in the final Office Action with the plain language of the claim that the limitations of “messages” and “business objects” are separate and distinct be applied instead.

*d. The Prior Art does not Teach or Suggest the Limitations of a Properly Interpreted Claim*

When the claims are interpreted correctly, the rejection based on the prior art in the Office Action is misapplied and irrelevant. The feature set forth in the independent claims of “verifying that a change to one instance of the common business object does not change the other copies of the common business object” is not taught in the references. Regardless of whether it “would be obvious . . . to modify the structure of Grewal by making sure that changes to one message do not change another message to ensure that incorrect message are not sent” as set forth in the final Office Action, this modification would not correspond with any properly construed limitation of the claim. Even if it was obvious from the prior art to make “sure that

changes to one message do not change another message,” there is no teaching or suggestion in the prior art that this modification related to messages be extended to a modification related to business objects as well. Business objects and messages are separate limitations, provide separate purposes, and otherwise related differently to the other limitations of the claims. There is no suggestion in the prior art that the prior art that the proposed modification take the extra necessary step to make obvious the properly construed limitation at issue in the claims.

*e. Conclusion*

Accordingly, Appellants respectfully submit that that the prior art does not teach or suggest a corresponding feature of the correctly interpreted claim limitations related to the former feature, and thus request that the Board of Appeals find that the claims are patentably distinguishable from the prior art of record.

2. The Rejection Related to the Feature of Batching the Files into Logically Related Requests

*a. The Rejection of the Feature as set forth in the Final Office Action*

The final Office Action also states on page 2 that Grewal does not “teach batching the files into logically related requests so as to send a single network message with a parent batch and a dependent batch.” Instead, the final Office Action continues on page 3 that “Nordstrom discloses an interrupt and message batching method (title), that would inherently have a parent batch and a dependent batch as Nordstrom discloses batching messages together (claim1).” The final Office Action also states on page 3 that “[i]t would have been obvious . . . to modify the system as disclosed by Grewal by batching messages as Grewal discloses holding messages in a queue until a number threshold is reached (abstract).” In support of the weight of the Grewal and Nordstrom references used in the rejection, the final Office Action states in the Response to Arguments section on page 4 that:

“As the instant application places an emphasis on ‘batching’ it is considered that the Grewal reference, which has an emphasis on balancing a traffic load of

messages, would be analogous. As to the inclusion of the Nordstrom reference, it is merely used to teach that batching messages is old and well known in the art.”

Appellants submit that the corresponding claim limitations require more than merely “batching,” and that the references of Grewal and Nordstrom do not teach or suggest the features of the claims.

*b. The Rejection does not Adequately Address the Limitations of the Claims*

The limitations at issue here are from element (a) of the independent claims, and exemplary claim 1 provides:

“(a) batching logically related requests for reducing network traffic, including the steps of managing a group of business objects necessary for a transaction in a logical unit of work, and grouping the logically related requests received from the logical unit of work into a single network message, wherein the logically related requests include at least a dependent batched request and a parent batched request.”

As can clearly be determined from the plain language of element (a) of the independent claims, the limitations set forth more than simply “batching messages together” as taught by Nordstrom or “holding messages in a queue until a number threshold is reached” as taught by Grewal and stated in the final Office Action.

For example, there is nothing to indicate that the prior art suggests that Grewal or Nordstrom be modified to include the step of “managing a group of business objects necessary for a transaction in a logical unit of work” as related to the batching or otherwise. Included in the argument above related to the former feature, the final Office Action cannot interpret “business objects” as “messages” in the way the final Office Action attempts to do so. Neither Grewal nor Nordstrom teach or suggest a counterpart feature of “business objects” that would correspond with the other claim elements in the manner set forth in the claim. Because this feature is neither taught nor suggest in the prior art of record, it cannot be found in any proposed combination of references.



Additionally, the final Office Action cannot support the assertion that batching messages together, as shown in Nordstrom, means that the method “would inherently have a parent batch and a dependent batch.” The Nordstrom reference’s disclosure of batching messages teaches batching that certainly could be considered separate from parent and dependent batching of logically related requests into a single network messages, as would be required if the teaching of Nordstrom was indeed “inherent.”

Nordstrom and Grewal do not even suggest a unique relationship between the logically related requests such as the parent/dependent batching set forth in the independent claims. Specifically, there is no teaching or suggestion in the prior art to address the limitations of element (b) of the independent claims including the steps related to “indicating whether the dependent batch request depends on the response to the parent batched request” and limitations of element (c) of the independent claims of “sending a single network message to the group of business objects necessary for the logical unit of work.”

c. Conclusion

Accordingly, Appellants respectfully submit that the final Office Action does not adequately address the limitations of the claims in the rejection, and that prior art otherwise does not teach or suggest the limitation related to the parent/dependent batching. Appellants respectfully request that the Board of Appeals find that the claims are patentably distinguishable from the prior art of record.

**(8) CLAIMS APPENDIX**

An appendix containing a copy of the claims 1-18, i.e., the claims involved in this appeal, is included with this Appeal Brief.

**(9) EVIDENCE APPENDIX**

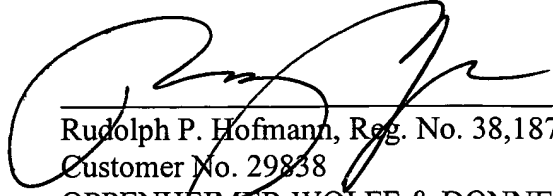
No evidence under 37 CFR 1.130, 1.131, or 1.132 or any other Office-entered evidence is relied upon in this appeal. Accordingly, no evidence appendix is included with this Appeal Brief.

**(10) RELATED PROCEEDINGS APPENDIX**

No decisions on related appeals have been rendered. Accordingly, no related proceedings appendix is included with this Appeal Brief.

**(11) CONCLUSION**

Pending claims 1-18 remain rejected under 35 U.S.C. 103(a). Appellants respectfully assert that the Office has not established a *prima facie* case of obviousness as demonstrated above. Appellants respectfully request that the Board of Patent Appeals and Interferences reverse the rejection.



---

Rudolph P. Hofmann, Reg. No. 38,187  
Customer No. 29838  
OPPENHEIMER WOLFF & DONNELLY LLP  
45 South Seventh Street, Suite 3300  
Minneapolis, Minnesota 55402  
Telephone: (612) 607-7340

## CLAIMS APPENDIX

1. (Previously presented) A method for implementing transaction services patterns, comprising the steps of:
  - (a) batching logically related requests for reducing network traffic, including the steps of managing a group of business objects necessary for a transaction in a logical unit of work, and grouping the logically related requests received from the logical unit of work into a single network message, wherein the logically related requests include at least a dependent batched request and a parent batched request;
  - (b) indicating whether the dependent batched request depends on the response to the parent batched request, including the steps of receiving a register that the dependent batched request is dependent upon response data from the parent batched request, receiving a response to the parent request, directing data from the response to the parent request to the dependent batched request; and receiving a response to the dependent batched request based on the response to the parent request;
  - (c) sending the single network message to the group of business objects necessary for the logical unit of work;
  - (d) sorting the logically related requests that are unbatched from a batched message; and
  - (e) providing multiple logical units of work operating concurrently, wherein the logical unit of work is one of the multiple logical units of work, such that each of the multiple logical units of work manipulates at least one of the group of business objects that is common to each of the multiple logical units of work, including the steps of creating a copy of the common business object for each of the logical units of work such that the copy of the common business object for each of the logical units of work is a separate instance of the common business object, and verifying that a change to one instance of the common business object does not change the other copies of the common business object.

2. (Previously presented) A method as recited in claim 1, wherein the step of batching logically related requests further includes the steps of:  
  
providing the group of business objects necessary for the transaction;  
  
storing the single network message; and  
  
sending the single network message upon receiving an order to send the message.
3. (Previously presented) A method as recited in claim 1, wherein the step of indicating whether the dependent batched request depends on the response to the parent batched request further includes the steps of:  
  
providing the group of business objects necessary for the transaction;  
  
sending the single network message across a network; and  
  
unbundling the requests from the network message.
4. (Previously presented) A method as recited in claim 1, wherein the step of sending the single network message to the group of business objects includes the steps of:  
  
providing the group of business objects necessary for the transaction;  
  
creating a receiver which communicates with the business objects in the logical unit of work;  
  
receiving a message for the business objects in the logical unit of work; and  
  
directing the message to the receiver,  
  
wherein the receiver forwards the message to each of the business objects in the logical unit of work.

5. (Previously presented) A method as recited in claim 1, wherein the step of sorting the logically related requests that are unbatched from a batched message includes the steps of:  
  
providing the group of business objects necessary for the transaction;  
  
obtaining at least one of sorting rules and sort weights;  
  
sorting the requests in the message and placing them in a specific order determined from the one of the sorting rules and the sort weights;  
  
batching the sorted requests into a single message;  
  
sending the message to a data server; and  
  
unbundling the requests from the message in the specific order.
6. (Previously presented) A method as recited in claim 1, wherein the step of providing multiple logical units of work operating concurrently further includes the steps of:  
  
receiving a request to make changes to a copy of the business object of one of the logical units of work and changing that copy of the business object, wherein each copy of the business object knows the context of that copy of the business object in relation to the associated logical unit of work and wherein the other copies of the business object are not changed; and  
  
updating the common business object based on the change to the copy of the business object.
7. (Previously presented) A computer program embodied on a computer readable medium for implementing transaction services patterns, comprising:
  - (a) a code segment that batches logically related requests for reducing network traffic, including a code segment that manages a group of business objects necessary for a



transaction in a logical unit of work, and a code segment that groups the logically related requests received from the logical unit of work into a single network message, wherein the logically related requests include at least a dependent batched request and a parent batched request;

- (b) a code segment that indicates whether the dependent batched request depends on the response to the parent batched request, including a code segment that receives a register that the dependent batched request is dependent upon response data from the parent batched request, a code segment that receives a response to the parent request, a code segment that directs data from the response to the parent request to the dependent batched request; and a code segment that receives a response to the dependent batched request based on the response to the parent request;
  - (c) a code segment that sends the single network message to the group of business objects necessary for the logical unit of work;
  - (d) a code segment that sorts the logically related requests that are unbatched from a batched message; and
  - (e) a code segment that provides multiple logical units of work operating concurrently, wherein the logical unit of work is one of the multiple logical units of work, such that each of the multiple logical units of work manipulates at least one of the group of business objects that is common to each of the multiple logical units of work, including a code segment that creates a copy of the common business object for each of the logical units of work such that the copy of the common business object for each of the logical units of work is a separate instance of the common business object, and a code segment that verifies that a change to one instance of the common business object does not change the other copies of the common business object.
8. (Previously presented) A computer program as recited in claim 7, wherein the code segment that batches logically related requests further includes:

- a code segment that provides the group of business objects necessary for the transaction;
  - a code segment that stores the single network message; and
  - a code segment that sends the single network message upon receiving an order to send the message.
9. (Previously presented) A computer program as recited in claim 7, wherein the code segment that indicates whether the dependent batched request depends on the response to the parent batched request further includes:
- a code segment that provides the group of business objects necessary for the transaction;
  - a code segment that sends the single network message across a network; and
  - a code segment that unbundles the requests from the network message.
10. (Previously presented) A computer program as recited in claim 7, wherein the code segment that sends the single network message to the group of business objects includes:
- a code segment that provides the group of business objects necessary for the transaction;
  - a code segment that creates a receiver which communicates with the business objects in the logical unit of work;
  - a code segment that receives a message for the business objects in the logical unit of work; and
  - a code segment that directs the message to the receiver,
- wherein the receiver forwards the message to each of the business objects in the logical unit of work.

11. (Previously presented) A computer program as recited in claim 7, wherein the code segment that sorts the logically related requests that are unbatched from a batched message includes:
  - a code segment that provides the group of business objects necessary for the transaction;
  - a code segment that obtains at least one of sorting rules and sort weights;
  - a code segment that sorts the requests in the message and placing them in a specific order determined from the one of the sorting rules and the sort weights;
  - a code segment that batches the sorted requests into a single message;
  - a code segment that sends the message to a data server; and
  - a code segment that unbundles the requests from the message in the specific order.
12. (Previously presented) A computer program as recited in claim 7, wherein the code segment that provides multiple logical units of work operating concurrently further includes:
  - a code segment that receives a request to make changes to a copy of the business object of one of the logical units of work and changing that copy of the business object, wherein each copy of the business object knows the context of that copy of the business object in relation to the associated logical unit of work and wherein the other copies of the business object are not changed; and
  - a code segment that updates the common business object based on the change to the copy of the business object.
13. (Previously presented) A computer-readable storage medium containing a set of instructions for a general purpose computer for implementing transaction services patterns, the set of instructions comprising:

- (a) means for batching logically related requests for reducing network traffic, including means for managing a group of business objects necessary for a transaction in a logical unit of work, and means for grouping the logically related requests received from the logical unit of work into a single network message, wherein the logically related requests include at least a dependent batched request and a parent batched request;
- (b) means for indicating whether the dependent batched request depends on the response to the parent batched request, including means for receiving a register that the dependent batched request is dependent upon response data from the parent batched request, means for receiving a response to the parent request, means for directing data from the response to the parent request to the dependent batched request; and means for receiving a response to the dependent batched request based on the response to the parent request;
- (c) means for sending the single network message to the group of business objects necessary for the logical unit of work;
- (d) means for sorting the logically related requests that are unbatched from a batched message; and
- (e) means for providing multiple logical units of work operating concurrently, wherein the logical unit of work is one of the multiple logical units of work, such that each of the multiple logical units of work manipulates at least one of the group of business objects that is common to each of the multiple logical units of work, including means for creating a copy of the common business object for each of the logical units of work such that the copy of the common business object for each of the logical units of work is a separate instance of the common business object, and means for verifying that a change to one instance of the common business object does not change the other copies of the common business object.

14. (Previously presented) A computer-readable storage medium as recited in claim 13, wherein the means for batching logically related requests further includes:
- means for providing the group of business objects necessary for the transaction;
- means for storing the single network message; and
- means for sending the single network message upon receiving an order to send the message.
15. (Previously presented) A computer-readable storage medium as recited in claim 13, wherein the means for indicating whether the dependent batched request depends on the response to the parent batched request further includes:
- means for providing the group of business objects necessary for the transaction;
- means for sending the single network message across a network; and
- means for unbundling the requests from the network message.
16. (Previously presented) A computer-readable storage medium as recited in claim 13, wherein the means for sending the single network message to the group of business objects includes:
- means for providing the group of business objects necessary for the transaction;
- means for creating a receiver which communicates with the business objects in the logical unit of work;
- means for receiving a message for the business objects in the logical unit of work; and
- means for directing the message to the receiver, wherein the receiver forwards the message to each of the business objects in the logical unit of work.

17. (Previously presented) A computer-readable storage medium as recited in claim 13, wherein the means for sorting the logically related requests that are unbatched from a batched message includes:

means for providing the group of business objects necessary for the transaction;

means for obtaining at least one of sorting rules and sort weights;

means for sorting the requests in the message and placing them in a specific order determined from the one of the sorting rules and the sort weights;

means for batching the sorted requests into a single message;

means for sending the message to a data server; and

means for unbundling the requests from the message in the specific order.

18. (Previously presented) A computer-readable storage medium as recited in claim 13, wherein the means for providing multiple logical units of work operating concurrently further includes:

means for receiving a request to make changes to a copy of the business object of one of the logical units of work and changing that copy of the business object, wherein each copy of the business object knows the context of that copy of the business object in relation to the associated logical unit of work and wherein the other copies of the business object are not changed; and

means for updating the common business object based on the change to the copy of the business object.